

## **REMARKS**

**[0002]** Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1, 3, 5-7, 10-11, 13, 15-17, 20-21, 23, 25-27, and 30-33 are presently pending. Claims 1, 3, 5-7, 11, 13, 15-17, 21, 23, 25-27, and 30 are amended herein. Claims 2, 4, 8-9, 12, 14, 18-19, 22, 24, and 28-29 are cancelled herein without prejudice or disclaimer. Claims 31-33 are herein added.

### **Formal Request for an Interview**

**[0003]** If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

**[0004]** Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

### **Claim Amendments and Additions**

**[0005]** Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 3, 5-7, 11, 13, 15-17, 21, 23, 25-27, and 30 herein. Claims 2, 4, 8-9, 12, 14, 18-19, 22, 24, 28-29 are cancelled herein without prejudice or disclaimer. Claims 31-33 are herein added.

**[0006]** Claim 1 is amended to recite “[a] method for handling a large data object in a database system implemented in a computer system...” Support for the amendment can be found throughout the Application including, for example, Field of the Invention Section where “[t]he present invention generally relates to the field of database systems and...” (Specification at paragraph [0001]).

**[0007]** Claim 1 is also amended to recite, *inter alia*, “[c]reating a handling structure...wherein partial update of the large data object referenced by the handling structure is processed without incurring substantial negative impact on overall performance of the database system.” Support for the amendment can be found throughout the Application including, for example, where in copying of large-value object using conventional technology, “[t]he overhead costs are much more significant and can negatively impact overall performance” when compared to copying many small-value objects (Specification at paragraph [0002]). The instant Application describes “[a] large object infrastructure where users/programmers can handle large values (data blocks) in the same way smaller values are handled,” thus diminishing or eliminating the substantial negative impact on the overall performance of the database system due to the heavy overhead costs derived from the copy of large-value objects. Moreover, it’s indicated in the Application that “[f]or various embodiments, the BH infrastructure may also support “partial updates” to data blocks--that is, the ability to change only a portion of a large value without incurring a full copy of the entire data value--to preclude some of the necessity for copying the entire data block...” (Specification at paragraph [0031]).

**[0008]** Independent claims 11 and 21 are amended to incorporate similar features, and therefore are also supported by the Application.

**[0009]** Similarly, claims 3, 13, and 23, which recite that “[t]he partial update of the large data object comprises replacing only a portion of the large data object without updating the large data object in its entirety,” are also supported by the Application. (Specification at paragraph [0031]).

**[0010]** Furthermore, Applicant adds new claims 31-33 herein. Support for the added claims can be found in the previously cancelled original claims 9, 19, and 29. In addition, claims 31-33 are amended to recite that the “lifetime [being] indicative of a length of time during which said handling structure is valid.” Support for the amendment can also be found in the Application. (Specification at paragraph [0042], i.e., “[s]everal embodiments utilize a “lifetime” property for the length of time a BH reference is valid”).

**[0011]** Accordingly, no new matter will be added by the paper. Entry to the file is respectfully requested.

## **Substantive Matters**

### **Claim Rejections under §§ 102 and/or 103**

**[0012]** Claims 1-30 are rejected under 35 U.S.C. § 102 and § 103 for being unpatentable over U.S. Patent No. 6,615,219 to Broso (“Broso”) and/in view of U.S. Patent No. 6,301,579 to Becker (“Becker”). Applicant respectfully traverses the rejections. In light of the amendments presented herein, Applicant submits that these rejections are moot.

**[0013]** Independent claim 1, as amended, recites (Emphasis added):

1. A method for handling a large data object in a database system implemented in a computer system, said method comprising:

creating a handling structure comprising at least a reference to locate the large data object stored in the database system and information to return an interface to provide access to the large data object in the database system;

wherein *partial update of the large data object referenced by the handling structure is processed without incurring substantial negative impact on overall performance of the database system.*

**[0014]** In making out the rejections to claim 1, the Office took the position that “Bruso discloses in figure 6, a method, computer-readable medium, and a system utilizing the same functionality for handling a large data object in a computer system, said method comprising creating a handling structure comprising a reference to locate the large data object and information to return an interface to provide access to the large data object (col.1, lines 59-67; col. 7, lines 42-49)...wherein said handling structure can be

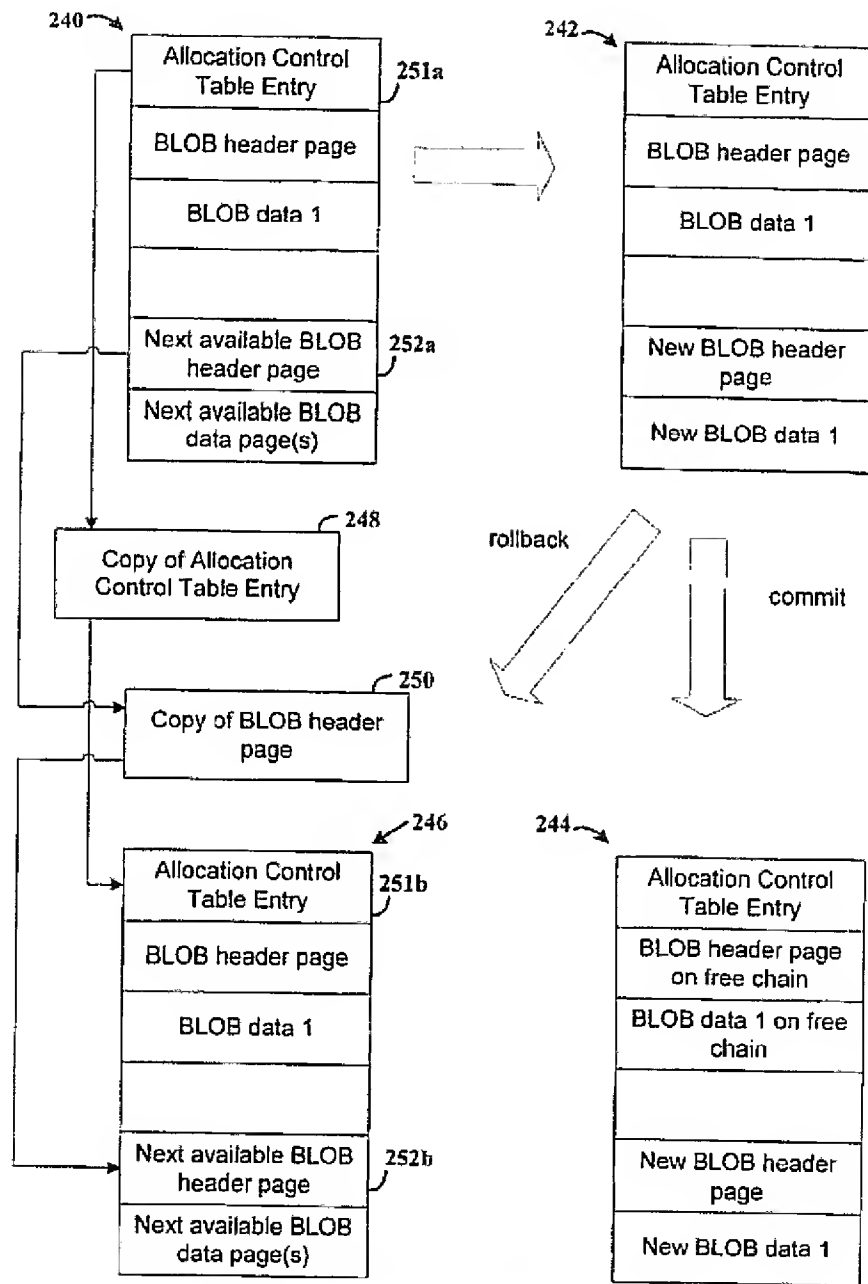
processed by said computer system, via functions(col. 6, lines 22-39), operations (col. 6, lines 50-59), and so forth available for a small data object (col. 3, lines 12-19), with which said large data object could not be so processed (col. 6, lines 19-22).” (Office Action 09/26/2007 at p.2).

**[0015]** Applicant respectfully traverses the rejections to claim 1. In particular, Applicant submits that Bruso does **NOT** teach a handling structure that “can be processed ...via functions, operations. and so forth available for a small data object, with which said large data object could not be so processed...” (Emphasis added).

**[0016]** Bruso is directed to a system and method for “[managing] binary large objects in a database [wherein] a database table includes a plurality of rows of data, [each of which] includes one or more fixed-length data elements and one or more object identifiers that reference and are associated with respective binary large object. An object handler coupled to the database management system, the object handler configured and arranged to store each binary large object in a section of contiguous storage referenced by the associated identifier...” (Bruso, Abstract).

**[0017]** In particular, Bruso teaches, *inter alia*, how the system supports rollback of changes to a database made by a transaction. To ensure atomicity, consistency, transaction isolation, and durability properties of a transaction that is fully supported in a DBMS (Database Management System), Bruso teaches a method to maintain atomicity of a transaction where the DBMS makes a copy of the data base page before making any transactions. This copy of the data page is called a “before look” copy. If the transaction is rolled back, the DBMS has a copy of the original contents of the page which it uses to restore the page to its original state. (Bruso, col.6, lines 12-19).

**[0018]** Since making a copy of pages containing a large size of BLOB requires a prohibitively long time to accomplish, Brusco teaches a method to efficiently implement the “before look” copy for the large size of BLOB, which is discussed in view of Fig. 4B. (Reproduced below).



**FIG. 4B**

**[0019]** Fig. 4B illustrates a diagram that shows different states of a sample portion of the DBMS in a transaction. In particular, block 240 (the initial state of a database prior to performing a delete request), block 242 (an intermediate state), block 244 (a first new state when a commit is requested by the transaction, indicating that the transaction is confirmed), and block 246 (a second new state when a rollback is requested by the transaction, indicating that the transaction is cancelled) are shown in Fig. 4B in response to a delete transaction.

**[0020]** According to Bruso, “[i]n processing a delete request, BLOB handler 306 links the BLOB header page into the free chain for the allocation control table from which the space was originally allocated. It sets the step-ID code in the BLOB header page to be the step-ID of the transaction which performed the delete operation. Writing the step-ID to the BLOB header page allows the BLOB handler to avoid reallocating the space to the same transaction until after the transaction makes a commit or rollback request which, in turn allows the BLOB handler to avoid making a copy of the unmodified BLOB pages (i.e., no before look of the page is required). The DBMS only makes a before look copy 248 of allocation control table entry 251a and a before look copy 250 of the BLOB header page 252a.” (Bruso, col. 6, lines 24-39 with Emphasis).

**[0021]** As illustrated in Bruso, “[i]f the transaction requests a commit, the copy of the allocation control table entry 248 and the BLOB header page 250 are discarded and the database updates are made permanent as shown by block 244... If the transaction request a rollback, **in the general case, the DBMS replaces all updated pages with the before look copies of the pages, as shown by block 246.**” (Bruso, col.6, lines 42-49 with Emphasis). Bruso then provides details of how the DBMS performs when BLOB is

involved. In particular, “[w]hen an insert operation” is involved before rollback, Bruso teaches to free up the space that had been used by the new BLOB data by “[w]riting the copy 248 of the allocation control table entry back to the file at 251b...” In an event that the BLOB header page came from a free chain in order to process the “insert operation”, a rollback to the initial status will include freeing the new BLOB header page and “[p]ut the BLOB header page back into the free chain.” (Bruso, col.6, lines 47-59).

**[0022]** In light of the detailed disclosure in Bruso above, Applicant respectfully submits that Bruso does not teach the feature “wherein said handling structure can be processed by said computer system, via functions, operations, and so forth available for a small data object, with which said large data object could not be so processed” recited in previously amended claim 1.

**[0023]** Nonetheless, in view of the amendment to claim 1, Applicant respectfully submits that Bruso does **NOT** teach the amended feature “partial update of the large data object referenced by the handling structure is processed without incurring substantial negative impact on overall performance of the database system” (Emphasis added).

**[0024]** As elaborated in paragraphs [0018]-[0021] above, Bruso at best teaches a “before look” copy without copying an entire BLOB to avoid prohibitive long time and a method to free up space that was used to store new BLOB data in response to an “insertion transaction” when a rollback command is requested. Bruso, in view of Figs. 5B, 7, and 8, teaches how a DBMS manages “inserting” and “deleting” a BLOB.

**[0025]** However, Bruso is completely silent in addressing an operation of a “partial update of BLOB” that does not “[incur] substantial negative impact on overall performance of the database system.” Particularly, the amended feature “partial update



of the large data object [comprising] replacing only a portion of the large data object without updating the large data object in its entirety” recited in claim 3 is absent in Bruso.

**[0026]** Accordingly, amended claims 1 and 3 are asserted patentably distinct from Bruso. Claims 11, 21, and 13, 23 are amended to incorporate features similar to those discussed above with references to claims 1 and 3. Accordingly, Applicant submits that these claims are also patentably distinct from Bruso for at least similar reasons.

**[0027]** In addition to the reasons mentioned above, Applicant respectfully submits that newly added claim 31 is patentably distinct from Bruso for the reasons discussed below.

**[0028]** The newly added claim 31 recites (Emphasis added):

31. The method of claim 1 wherein said handling structure has a lifetime *indicative of a length of time during which said handling structure is valid*, and said handling structure further comprises a field having a value corresponding to said lifetime.

**[0029]** In rejecting the previously presented claim 1 “[w]herein said handling structure has a lifetime, and the said handling structure comprising a field having a value corresponding to said lifetime...,” the Office referred to the “timestamp” in Bruso.

**[0030]** Applicant respectfully traverses the rejection. Claim 31 is derived from previously presented claim 1, and is rewritten from previously cancelled claim 9. In view of the amendment, Applicant submits that the feature “*lifetime indicative of a length*

*of time during which said handling structure is valid*” is absent in Bruso. (Emphasis added).

**[0031]** The “timestamp” as illustrated in Bruso, refers to “[t]he time at which the BLOB was written to the storage area and has a precision level of nanoseconds...The creation time of the row must match the creation time of the corresponding BLOB.” (Broso, col.4, lines 29-35).

**[0032]** Applicant submits that the “lifetime” recited in claim 31 is not the “timestamp” in Bruso. Instead, the “lifetime [indicates] the length of time during which said handling structure is valid” (Claim 31). Accordingly, the feature “lifetime” is absent in Bruso.

**[0033]** Since claims 32 and 33 incorporate the similar feature, these claims are asserted patentably distinct from Bruso for the reasons referenced to claim 31.

**[0034]** Furthermore, in addition to the reasons in paragraphs [0017]-[0026], claim 7 is asserted patentably distinct from Bruso for the following reason.

**[0035]** Claim 7, as amended, recites (Emphasis added):

7. The method of claim 1 wherein *the reference of said handling structure is configured to point to a small value data object within the handling structure itself provided that said small value data object is stored entirely within said handling structure.*

**[0036]** In rejecting claim 7, the Office refers to Bruso wherein “[a] significant portion of today’s database technology was built when the data items in a database were relatively small. Smaller data items permitted multiple rows of data to be stored in one

physical page of storage, thereby enabling retrieval of multiple rows with a single input/output (I/O) operation...” (Bruso, col.1, lines 30-35).

**[0037]** Applicant respectfully traverses the rejection. In particular, “[s]maller data items [permitting] multiple rows of data to be stored in one physical page of storage thereby enabling retrieval of multiple rows with a single input/output,” as illustrated in Bruso, does not provide a sufficient support for the Office to make the assertion in the Office Action that Bruso teaches the emphasized feature in claim 7.

**[0038]** In view of the context of claim 1 on which claim 7 depends, Applicant herein submits that the handling structure comprising a reference “to locate the large data object stored in the database” as well as “to a small value data object within the handling structure itself provided that the small value data object is stored entirely within the handling structure” is absent in Bruso.

**[0039]** Accordingly, since claims 17 and 27 are amended to incorporate at least the similar feature, these claims are asserted patentably distinct from Bruso for the similar reasons referenced to claim 7.

### **Dependent Claims**

**[0040]** In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

## CONCLUSION

**[0041]** All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the Examiner is urged to contact me before issuing a subsequent Action. Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Dated: 2008-03-26

By: Ningning Xu

Ningning Xu  
Reg. No. L0293  
(509) 324-9256 x226  
[ningning@leehayes.com](mailto:ningning@leehayes.com)  
[www.leehayes.com](http://www.leehayes.com)

My Assistant: Carly Bokarica  
(509) 324-9256 x264  
[carly@leehayes.com](mailto:carly@leehayes.com)